# Aha Moment Revisited: Are VLMs Truly Capable of Self Verification in Inference-time Scaling?

Anonymous ACL submission

## Abstract

Recent advances in large language models (LLMs) have demonstrated that inference-time computation techniques, such as decoding-time scaling and self-refinement, can significantly enhance reasoning capabilities without relying on external knowledge. A key driver of this success is the emergence of self-correction and self-verification behaviors, often elicited through reinforcement learning (RL).

In this paper, we investigate whether these inference-time techniques extend effectively to vision-language models (VLMs), particularly those trained with RL. We find that while decoding strategies such as majority voting and best-of-N selection with self-verification all improve VLM reasoning performance, generation-reliant methods such as the former achieve significantly higher gains versus verification-reliant methods such as the latter. Additionally, the self-correction behavior often associated with RL-tuned models, such as “aha moment,” does not lead to measurable gains. We show via extensive experimentation within the inference-time scaling framework to identify a key root cause: RL-trained VLMs still lack robust self-verification capabilities across both visual and textual modalities.

## 1 Introduction

The reasoning capabilities of large language models (LLMs) have seen notable improvements in recent years (DeepSeek-AI, 2025; OpenAI, 2024). Although larger model scales and higher-quality pretraining datasets are major contributing factors to these improvements, emerging strategies that instead leverage **inference-time computation** (Snell et al., 2024) have also been proven effective: Providing models with zero-shot "think step by step" prompts or few-shot demonstrations augmented with intermediate reasoning steps (Wei et al., 2022) have enabled generation of extended reasoning chains even when not explicitly fine-tuned to do so. Likewise, methods such as decoding-time majority vote (Wang et al., 2023) and chain-of-thought decoding (Wang and Zhou, 2024) have enabled outputting of higher-quality answers without external feedback. More recently, inference-time self-correction (Kumar et al., 2025; DeepSeek-AI, 2025) has emerged as another form of scaling: models are trained with Reinforcement Learning (RL) to revise earlier mistakes and generate additional reasoning steps to arrive at improved reasoning answers. "**aha moment**" exists: model generates "Wait, I made a mistake in my prior response”—and initiates a second round of reasoning to refine its answer.

A dominant hypothesis for why inference-time computation works without external knowledge is that models contain difficult-to-access “hidden knowledge” (Huang et al., 2024; Hinton et al., 2015), and that these prompting and/or decoding methods, rather than being knowledge generators on their own, serve as effective extractors of hidden knowledge for further reasoning into more user-accessible forms.

**What exactly is this hidden knowledge?** A compelling possibility is that it is the models’ capacities for **inference-time self-verification**. The various aforementioned methods invoke different degrees of self-verification, from zero-shot "think step by step"’s implied verification against an answer template to more explicit verification present within RL-trained, "aha-moment" utilizing models. LLM-Monkey (Brown et al., 2024) demonstrates that with sufficiently powerful verification capabilities, one can simply sample multiple diverse outputs from the model and select the most accurate one to improve performance (Song et al., 2025). Interestingly, Song et al. (2025) shows that LLMs often perform even better on verifying answers versus generating them: this gap may explain why inference-time computation methods which invoke explicit self-verification such as Self-

Refine (Madaan et al., 2023) achieve high effectiveness in LLM reasoning.

A central question we explore in this work is whether **self-verification generalize swell to VLMs.** Notably, several recent efforts (Zhou et al., 2025; Chen et al., 2025b; Zhang et al., 2025; Huang et al., 2025; Liu et al., 2025; Deng et al., 2025; Wang et al., 2025) adopt similar RL-based training strategies and report the emergence of “aha moments” in VLM reasoning to suggest that VLMs similarly contain the "hidden knowledge" of self-verification capacitiy present in LLMs and can be elicited via RL. However, a key question remains: **Are RL-trained VLMs genuinely effectively performing self-verification and self-correction during inference, or are these behaviors merely surface-level artifacts of training which contribute little to model performance?**

We study this problem by contrasting two inference-time strategies: (1) **Majority vote**, which generates multiple answers, then determines the final answer via a consensus among the generated answers. This method does focus on self-verification, instead requiring a model to have high generation capabilities for consistently outputting correct answers. (2) **Self-verified Best-of-N**, which similarly generates multiple answers, but explicitly uses itself as a verifier to evaluate and select the most appropriate self-generated answer, placing heavy emphasis on the model’s verification capability for performance. Importantly, we find that the former approach consistently outperforms the latter for a variety of evaluated RL-trained VLMs, highlighting a notable presence of a generation-verification gap present in VLMs but absent in LLMs.

Finally, we probe the self-verification mechanism in more detail to study possible causes of this gap by comparing between (1) giving the (self-)verifier access to the original image input and (2) witholding it during verification within the best-of-N setup, and find that the verifier counterintuitively performs better **without the image input**. This behavior highlights a possible core limitation of VLMs’ self-verification, in that they currently do not sufficiently leverage visual information for self-verification, which may explain the fundamental limitations that prevent current VLMs to effectively performing inference-time computation.

**Contributions.** In this paper, we explicitly demonstrate that inference-time decoding strategies improve reasoning performance in RL-tuned VLMs. We also show that the emergence of “aha moments” in RL-tuned VLMs does not lead to gains in final reasoning accuracy—largely due to the model’s limited self-verification capabilities. We design and perform extensive experimentation with various inference-time scaling frameworks to support our findings.

## 2 Related Works

### 2.1 LLM/VLM, Reinforcement Learning for Reasoning

Reinforcement learning (RL) was introduced to LLM fine-tuning via RL from human feedback (RLHF) (Ouyang et al., 2022), which learns a reward model from human preferences and optimizes the LLM policy, using Proximal Policy Optimization (PPO) (Schulman et al., 2017). More recent works (Rafailov et al., 2023; Shao et al., 2024) are multiple variants of PPO with improved computational efficiency. Beyond alignment, RL has also been shown to enhance LLM reasoning and self-correction capabilities (Kumar et al., 2025; DeepSeek-AI, 2025; Zeng et al., 2025a). Several studies (Gandhi et al., 2025; Zeng et al., 2025a) further investigate what intrinsic properties enable effective self-improvement and how "aha moments" emerge as a result of RL-based training.

In the vision-language domain, similar ideas have been extended to improve VLM reasoning. A number of recent works apply RL to incentivize multimodal reasoning behaviors, typically using PPO or GRPO to fine-tune VLMs. These studies report positive signs of RL to train VLM to generate "aha moments" in VLMs (Zhou et al., 2025; Chen et al., 2025b; Zhang et al., 2025; Huang et al., 2025; Liu et al., 2025; Deng et al., 2025; Wang et al., 2025).

### 2.2 Inference-Time Scaling

Inference-time scaling (Snell et al., 2024; Brown et al., 2024) has emerged as an effective strategy for improving LLM reasoning without additional fine-tuning. Several methods fall under this umbrella. Simple parallel decoding approaches—such as chain-of-thought decoding (Wang and Zhou, 2024) and self-consistency sampling (Wang et al., 2023)—have shown strong empirical gains by aggregating multiple sampled outputs. More sophisticated techniques involve training reward-based verifiers to guide step-by-step generation (Lightman et al., 2023). Recent studies have also proposed training-time modifications to enhance inference-time behavior. For example, inference-aware fine-tuning methods (Chow et al., 2024; Qu et al., 2024) aim to improve best-of-N. Meanwhile, sequential refinement approaches—such as Think-Speak (Goyal et al., 2024)—encourage the model to iteratively revise its own answers in a sequential (rather than parallel) manner, offering a complementary view of inference-time reasoning.

## 3 Methodology

We investigate the impact of various inference-time scaling methods for VLMs, methods that are considered established methods for text-based-only LLMs. As we introduce these methods to VLMs, we analyze the adaptability of each method with respect to gains in performance and to evidence of the emergence of self-verification capabilities.

### 3.1 Inference-time Scaling Methods

#### 3.1.1 Reinforcement Learning for VLMs

We test both the base version as well as the RL-tuned version of the VLMs. Previous work on LLMs has demonstrated the emergence of the 'aha moment' in RL RL-tuned model's reasoning process and such a process was shown to have a positive contribution to model performance. We would like to study whether similar benefits also exist for VLMs trained through RL. To this end, we adopt RL-tuned models from recent work (Zhang et al., 2025; Chen et al., 2025a; Wang et al., 2025), using them directly within our experimental framework. These models are trained under a general RL objective commonly used for multimodal reasoning:

$$\max_{\pi_\theta} \mathbb{E}_{[I,x]\sim\mathcal{D},\, y\sim\pi_\theta(\cdot|I,x)}\big[r_\phi(I,x,y)\big] - \beta\,\mathbb{D}_{\mathrm{KL}}\big[\,\pi_\theta(\cdot \mid I,x) \,\|\, \pi_{\mathrm{ref}}(\cdot \mid I,x)\big]. \quad (1)$$

where $\pi_\theta$ is the policy VLM parametrized with model weights $\theta$. $\pi_{\text{ref}}$ is the reference VLM policy. $r_\phi$ is the reward function. $\mathbb{D}_{\text{KL}}$ is KL-divergence measure. $\beta > 0$ is the KL penalty coefficient. The input $[I, x]$ denotes multimodal samples with image and text drawn from the dataset $\mathcal{D}$. The generated response in the rollout $y \sim \pi_\theta$, sampled from the VLM policy. Specifically, inspired by DeepSeek-R1 (DeepSeek-AI, 2025), most recent RL-for-VLM work adopts *Group Relative Policy Optimization* (GRPO), which removes the need for a separate value-function critic by estimating a baseline directly from a *group* of sampled roll-outs, thereby cutting both memory usage and wall-clock time. For every multimodal prompt $[I, x]$, we first freeze the current policy to create a snapshot $\pi_{\text{old}}$. We then draw $G$ candidate outputs $\{y_i\}_{i=1}^{G} \sim \pi_{\text{old}}(\cdot \mid I, x)$ and compute token-level advantages $\hat{A}_{i,t}$ by subtracting the group-mean return from each candidate's return. The policy is updated by maximizing the clipped-surrogate objective equation 2

This GRPO objective, $\mathcal{J}_{\text{GRPO}}(\theta)$, aims to update the policy $\pi_\theta$ by maximizing an expected, clipped surrogate objective based on multiple candidate generations from an old policy $\pi_{\text{old}}$. The core term involves a probability ratio $r_{i,t}(\theta)$ between the current and old policies for each token, multiplied by a token-level advantage $\hat{A}_{i,t}$ (derived from comparing a candidate's return to the group mean). This product is clipped to limit policy update sizes, promoting stability, a technique common in PPO. A KL-divergence penalty term, $-\beta\,\mathbb{D}_{\text{KL}}[\pi_\theta \,\|\, \pi_{\text{ref}}]$, regularizes the policy $\pi_\theta$ to prevent it from straying too far from a reference policy $\pi_{\text{ref}}$.

These RL-tuned VLMs are typically optimized using outcome-based rewards, and recent works (Zhang et al., 2025; Chen et al., 2025a; Wang et al., 2025) claim near-GPT-4o-level performance using models with only ~7B parameters. They also report the emergence of "aha moments"—suggesting that the models can learn to self-correct by identifying failures in earlier reasoning and generating additional rethinking steps, which can be considered as emergent inference-time scaling behavior.

#### 3.1.2 Decoding Methods for VLMs

VLMs generate text in the same way as LLMs do, except with additional image embeddings as part of the input query. Decoding methods concerned with how each next token is sampled from Language Models. In this work, we consider methods that aim to sample multiple starting tokens and thus generate multiple outputs given one single input query.

**Greedy Decoding** Sequentially selects the most probable next token at each decoding step. It is a one-time inference with no scaling.

**Decoding-Time Majority Voting** This strategy first samples multiple candidate outputs and then subsequently selects the final solution by majority consensus among the generated candidates. By aggregating multiple responses, it seeks to mitigate random errors or inconsistencies in individual outputs. We consider this as a strong baseline method to beat due to the 'deterministic' nature of how the

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{[I,x]\sim\mathcal{D}\{y_i\}_{i=1}^{G}\sim\pi_{\text{old}}(\cdot|I,x)} \left[ \frac{1}{G}\sum_{i=1}^{G}\frac{1}{|y_i|}\sum_{t=1}^{|y_i|}\min\Big(r_{i,t}(\theta)\,\hat{A}_{i,t}, \text{clip}\big(r_{i,t}(\theta),\,1-\epsilon,\,1+\epsilon\big)\,\hat{A}_{i,t}\Big) \;-\; \beta\,\mathbb{D}_{\text{KL}}\big[\pi_\theta \,\|\, \pi_{\text{ref}}\big] \right] \tag{2}$$

final output is selected. **This method primarily reflects the model's generation capability, as high accuracy depends on the model producing correct answers frequently enough to dominate the vote.**

**Best of N Sampling with Self as Verifier** This strategy also samples multiple candidate outputs from the VLM, but we then prompt the model to evaluate and verify all candidate outputs together. The output identified as most reliable by the model itself is selected as the final answer, thus integrating a self-verification component into the decoding process. **This method emphasizes the model's self-verification ability, as accuracy depends on correctly identifying the best answer to the question from a diverse set of responses.**

**Chain-of-Thought(CoT) Decoding** Unlike Greedy Decoding, Chain-of-Thought(CoT) Decoding considers multiple candidate tokens (top-k) at critical decoding points, branching out to form multiple decoding paths. Each of these paths potentially includes intermediate reasoning steps generated inherently by the model. A distinguishing feature of CoT Decoding is the use of a confidence metric, computed as the average probability margin between the top two candidate tokens across answer tokens within each decoding path. The path exhibiting the highest confidence margin is selected as the final output.

**Verifier Prompt**

```
Now you act as a judge, helping me
determine which of the <length> texts I
provide better answers the question.
Question: <question>
Repsonse: <response>
Please strictly follow the following
format requirements when outputting, and
don't have any other unnecessary words.
Output format: "I choose response
[number] because"
```

### 3.2 Evaluating Decoding Methods

To quantify the effectiveness of each decoding strategy, we use reasoning accuracy as our primary evaluation metric. **Accuracy** is defined as the proportion of examples where the final selected answer exactly matches the ground-truth solution.

Importantly, we focus on accuracy rather than coverage—the latter referring to the percentage of examples where at least one generated candidate is correct—because we do not assume access to a strong oracle verifier. Instead, our goal is to assess whether the model can effectively self-verify and select the best answer on its own, thereby reflecting its true reasoning performance.

### 3.3 Self-Verification in Vision Language Models

Self-verification has emerged as an influential generation strategy within LLMs, enabling models to internally assess and validate the accuracy and reliability of their generated outputs. In our study, we investigate whether similar self-verification capabilities exist within VLMs. Utilizing the VLM itself as the verifier in the "Best of N" Decoding strategy allows for direct evaluation of the model's self-verification abilities.

The self-verification mechanism typically involves the model generating multiple candidate outputs and subsequently scoring or ranking these candidates based on internal measures of confidence, coherence, and contextual alignment. This intrinsic verification mechanism provides insights into the model's reflective reasoning capabilities—its capacity to recognize correct reasoning pathways and distinguish them from incorrect or less coherent alternatives.

To understand whether VLMs are able to benefit from self-verification and whether the vision inputs are been used for better self-verification, we test VLMs using multiple configurations of the Best of N Decoding method:

- **Self Verification with Text Only:** The self-verifier receives only the generated responses and the text-based question. The image is omitted to test the model's ability to verify using language alone.
- **Self Verification with Image and Text:** The self-verifier is provided with both the image and the text input, allowing it to use multi-modal information for verification.

### 3.4 Finding 'aha moment'

The 'aha moment' is being regarded as a signature of the self-verification process for LLMs. We investigate if such 'aha' moment is contributing meaningfully to VLMs.

**Aha Search Method**. To systematically examine whether these "aha moments" after RL contribute to improved reasoning, we adopt an automatic detection protocol based on the Aha Search strategy (Gandhi et al., 2025; Zeng et al., 2025b). Originally proposed for LLMs, this method aligns "aha moments" with observable cognitive behaviors—specifically, backtracking and verification. In our setup, we prompt GPT-4o with the generated response and ask whether it exhibits these behaviors. The simplified prompt template is provided below and complete version is in appendix. If GPT-4o confirms the presence of backtracking or verification, we classify the response as containing an "aha moment."

**AHA Search Prompt**

```
<system prompt>
<start_of_reasoning> <RESPONSE>
<end_of_reasoning>
Specifically, actively identify and
emphasize beneficial behaviors such as:
(1) Backtracking: Explicitly revising
approaches upon identifying errors or dead
ends ...
(2) Verification: Systematically checking
intermediate results or reasoning steps
...
Important:
Clearly specify each beneficial behavior
you identify.
If there is a strong example of this,
provide <YES> followed by specific
explanations. Otherwise, provide <NO>
<NO>
```

We introduce two metrics to assess whether the presence of an aha moment positively contributes to reasoning performance:

**Post-Aha Accuracy Among Selected Predictions:** We compute the probability that a selected answer containing a confirmed aha moment is also correct. This is denoted as

$$P^{\star}(\text{Correct} \mid \text{AHA in Prediction}),$$

where the star ($\star$) indicates that we report the best value across all decoding strategies. This metric reflects how often aha moments align with correct final answers in selected outputs.

**Aha Potential Recovery Rate from Incorrect Predictions:** To assess whether aha moments can help recover from initial errors, we focus on cases where the selected prediction is incorrect. We then search through the unselected generated responses and check whether any of them contain both a confirmed aha moment and a correct answer. This is measured as

$$P(\text{Aha Correct} \mid \text{Wrong Prediction}),$$

indicating the potential for aha-based reasoning paths in the inference-time scaling to correct mistakes even when they are not selected by default.

## 4 Experiment

### 4.1 Dataset

We utilize the GeoQA170K and MathVista (Lu et al., 2024) datasets (Gao et al., 2023) for our empirical evaluation.

GeoQA170K is a geometric reasoning ability training dataset containing question-answer pairs created by a variety of models. We filter out repeated Q-A pairs and use 754 unique samples for our experimentation. All questions are in the form of an image + text prompt, while expected answers are free-form text from which only numerical symbols are extracted for evaluation based on numerical matching.

MathVista (Lu et al., 2024) covers a broad spectrum of visual question answering tasks, encompassing geometric, algebraic, arithmetic, and other forms of reasoning. The questions are similarly in the form of image + text, with images consisting of both simple mathematical diagrams and complex, real-world images associated with the text prompt. We use the test-mini split of the dataset, which contains 1,000 samples. Answer formats include both free-form responses and multiple-choice selections. Due to the diversity of the former, MathVista utilizes a LLM judge (parser prompt in Appendix) whether a predicted answer matches the ground truth.

### 4.2 Inference Setup

Our experiments are conducted on one computer equipped with NVIDIA 4090Ti and one with NVIDIA A100 GPU. The models evaluated range from the base Qwen2-VL-2B-Instruct to a set of Qwen-based RL-tuned models. The full list includes: R1-VL-2B, R1-VL-7B (Zhang et al., 2025), VLAA-Thinker-Qwen2.5VL-3B, VLAA-Thinker-Qwen2.5VL-7B (Chen et al., 2025a), and VL-Rethinker-7B (Wang et al., 2025). For sampling-

Table 1: Decoding Comparison on GeoQA with ×4 Scaling

| | Greedy | BoN w. Image | BoN w/o Image | Majority Votes | Chain-of-Thought |
|---|---|---|---|---|---|
| Qwen2-VL-2B-Instruct | 13.8 | **16.3** | 15.6 | 16.0 | 15.5 |
| R1-VL-2B (Zhang et al., 2025) | 26.9 | 28.9 | 28.2 | 30.2 | **31.0** |
| R1-VL-7B (Zhang et al., 2025) | 39.7 | **44.6** | 43.9 | 44.2 | 43.4 |
| VLAA-Thinker-3B (Chen et al., 2025a) | 44.2 | 27.5 | 31.6 | **46.4** | 45.1 |
| VLAA-Thinker-7B (Chen et al., 2025a) | 48.3 | 44.3 | 46.2 | **52.1** | 49.7 |
| VL-Rethinker-7B (Wang et al., 2025) | 60.1 | 59.9 | 59.8 | **61.9** | 61.4 |

Table 2: Conditional Accuracy w.r.t A-ha Moments

| | $P^\star$(Correct \| A-ha in Prediction) | $P$(A-ha Correct \| Wrong Prediction) |
|---|---|---|
| R1-VL-2B (Zhang et al., 2025) | 28.1(CoT) | 2.7 |
| R1-VL-7B (Zhang et al., 2025) | 49.5(VLM) | 4.4 |
| VLAA-Thinker-3B (Chen et al., 2025a) | 48.4(CoT) | 5.4 |
| VLAA-Thinker-7B (Chen et al., 2025a) | 49.5(CoT) | 13.0 |
| VL-Rethinker-7B (Wang et al., 2025) | 65.5(Majority Vote) | 19.5 |

based decoding methods—including majority voting and best-of-N with self-verification—we use the following inference configuration: temperature = 0.6, top-k = 50, and top-p = 0.9. For chain-of-thought decoding, which follows a deterministic approach as greedy decoding, we set the temperature to 0. For evaluation tasks such as MathVista grading and Aha moment detection, we use GPT-4o-mini as the LLM-based judge. We fix the random seed across all experiments to ensure reproducibility.

### 4.3 Discussion

In this section, we present key insights from our study, supported by extensive experimental results under the inference-time scaling framework. We find that RL-trained VLMs do benefit from inference-time scaling via parallel decoding strategies such as majority voting. However, the effectiveness of sequential inference-time scaling—those that rely on self-correction capabilities, such as "aha moments"—is far less clear. Our results indicate that such self-correction behaviors do not meaningfully improve VLM reasoning.

We further investigate this limitation and offer a potential explanation: RL-trained VLMs struggle with self-verification. We provide two pieces of evidence to support this claim. First, generation-heavy strategies like majority voting consistently outperform verification-heavy approaches such as best-of-N sampling with self-verification. Second, and more surprisingly, the self-verifier performs better when the image input is omitted—suggesting that the model does not effectively use visual information during the verification process.

Together, our findings highlight a fundamental gap in current VLM capabilities and represent a first step toward understanding the limitations and potential of inference-time scaling in multimodal reasoning.

**Inference Time Scaling Improves Performance of VLM**. Table 1 summarizes the performance gains of various inference-time scaling techniques versus the baseline deterministic, greedy decoding on GeoQA of the various VLMs. Notably, both BoN-based methods achieve limited performance gains over the greedy baseline, and in the case of the two VLAA VLMs, even result in performance *decreases* (up to -16.7%), which can be attributed to its tendency to re-do the question rather than to judge the response despite being explicitly prompted to choose from the responses. On the other hand, the two generation-emphasizing methods—Majority vote and CoT—achieve more steady performance gains (4.5% and 4.1%, respectively).

**RL-trained VLMs Do Not Benefited from Aha Moments** As shown in Table 2, answers flagged as containing "aha moments" do not lead to higher accuracy—even when we select the best result across all decoding strategies. This suggests that **"aha moments" do not reliably contribute to improved reasoning**. While we also assess the potential of aha moments—i.e., whether they could correct an initially wrong prediction—the observed probabilities remain low. This indicates that simply encouraging aha behavior is insufficient for improving model performance within the inference-time scaling framework.

**Current RL-trained VLM Fall Short in Verifi-**

Table 3: Verifier Comparison on GeoQA

| GeoQA | BoN w. Image | BoN w/o Image | Majority Votes |
|---|---|---|---|
| Scaling × 4 | | | |
| R1-VL-2B (Zhang et al., 2025) | 28.9 | 28.2 | **30.2** |
| R1-VL-7B (Zhang et al., 2025) | **44.5** | 43.9 | 44.2 |
| VLAA-Thinker-3B (Chen et al., 2025a) | 27.5 | 31.6 | **46.4** |
| VLAA-Thinker-7B (Chen et al., 2025a) | 44.3 | 46.2 | **52.1** |
| VL-Rethinker-7B (Wang et al., 2025) | 59.9 | 59.8 | **61.9** |
| Scaling × 8 | | | |
| R1-VL-2B (Zhang et al., 2025) | 31.2 | 30.2 | **35.1** |
| R1-VL-7B (Zhang et al., 2025) | 45.8 | 46.2 | **46.9** |
| VLAA-Thinker-3B (Chen et al., 2025a) | 23.5 | 28.0 | **48.3** |
| VLAA-Thinker-7B (Chen et al., 2025a) | 52.3 | 52.9 | **57.7** |
| VL-Rethinker-7B (Wang et al., 2025) | 58.9 | 58.9 | **62.1** |

Table 4: Verifier Comparison on MathVista

| MathVista | BoN w. Image | BoN w/o Image | Majority Votes |
|---|---|---|---|
| Scaling × 4 | | | |
| R1-VL-2B (Zhang et al., 2025) | 39.2 | 40.9 | **52.7** |
| R1-VL-7B (Zhang et al., 2025) | 59.3 | 63.8 | **65.2** |
| VLAA-Thinker-3B (Chen et al., 2025a) | 50.3 | 52.1 | **66.2** |
| VLAA-Thinker-7B (Chen et al., 2025a) | 65.5 | 58.2 | **71.6** |
| VL-Rethinker-7B (Wang et al., 2025) | 75.0 | 74.7 | **75.4** |
| Scaling × 8 | | | |
| R1-VL-2B (Zhang et al., 2025) | 41.5 | 42.0 | **56.4** |
| R1-VL-7B (Zhang et al., 2025) | 61.1 | 63.6 | **66.0** |
| VLAA-Thinker-3B (Chen et al., 2025a) | 48.4 | 45.1 | **65.6** |
| VLAA-Thinker-7B (Chen et al., 2025a) | 70.5 | 66.2 | **74.0** |
| VL-Rethinker-7B (Wang et al., 2025) | 73.9 | 71.4 | **75.6** |

**cation in Inference-time Scaling** Tables 3 and 4 quantitatively assess verification ability using best-of-N decoding with self-verification. Across both 4- and 8-sample settings, majority voting—an indicator of generation quality—consistently outperforms self-verification. This stands in contrast to findings in the LLM literature, where verification is often easier than generation. Our results suggest that current RL techniques do not endow VLMs with strong self verification capabilities, raising concerns about their effectiveness in multimodal reasoning tasks.

**No Visual Verification** Another notable observation from Tables 3 and 4 is that RL-trained VLMs sometimes verify their own outputs more accurately when visual input is excluded. This is particularly evident in the GeoQA dataset, which consists entirely of geometric questions. Including the image does not necessarily help the model judge correctness—suggesting that the VLM fails to integrate visual context during self-verification. Instead, the model over-relies on textual input, rendering its verification process in both modalities unreliable. Our findings show that current VLMs do not fully utilize visual information during verification, and we call for future research to address this shortcoming by enhancing the model's true multimodal verification capabilities to improve reasoning performance.

## 5 Conclusion

In this paper, we investigated the extensibility of LLM inference-time computation techniques to VLMs. We find that current RL-trained VLMs yet lack robust self-verification capabilities across both visual and textual modalities in the form of a verification-generation gap. We have performed extensive experimentation to support this claim: our results show that the verification-reliant best-of-N selection strategy achieves lower performance gains versus the generation-reliant majority voting, and that the self-correction behavior often associated with RL-tuned models, such as "aha moment," does not lead to measurable gains.

**Broader Impacts**. The current trend in the community treats vision-language models (VLMs) as a natural extension of large language models (LLMs), with many efforts focused on directly transfer-

ring reasoning successes from LLMs to VLMs. However, this paper highlights a critical gap: the core mechanisms that drive reasoning improvements in LLMs—particularly those enabled by reinforcement learning—do not translate effectively to VLMs. We argue that a key reason for this is the lack of robust multimodal self-verification capabilities in current VLMs, which undermines the foundation upon which RL-based reasoning succeeds in the LLM setting.
**LLM Use**. We use LLM for grammer checks.

## 6 Limitations

This work empirically highlights a key limitation of RL-trained VLMs: despite improvements in reasoning performance, these models struggle to fully realize their potential due to weak self-verification capabilities in multimodal settings. **While we analyze and diagnose this issue, we do not propose a solution to address it**. Instead, our findings serve as an important stepping stone—calling for future research to better understand and enhance the unique challenges and opportunities in VLM self-verification, a capability that remains underexplored in the current landscape.

## References


Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. 2024. Large language monkeys: Scaling inference compute with repeated sampling. *Preprint*, arXiv:2407.21787.

Hardy Chen, Haoqin Tu, Fali Wang, Hui Liu, Xianfeng Tang, Xinya Du, Yuyin Zhou, and Cihang Xie. 2025a. Sft or rl? an early investigation into training r1-like reasoning large vision-language models. *Preprint*, arXiv:2504.11468.

Liang Chen, Lei Li, Haozhe Zhao, Yifan Song, and Vinci. 2025b. R1-v: Reinforcing super generalization ability in vision-language models with less than $3. `https://github.com/Deep-Agent/R1-V`. Accessed: 2025-02-02.

Yinlam Chow, Guy Tennenholtz, Izzeddin Gur, Vincent Zhuang, Bo Dai, Sridhar Thiagarajan, Craig Boutilier, Rishabh Agarwal, Aviral Kumar, and Aleksandra Faust. 2024. Inference-aware fine-tuning for best-of-n sampling in large language models. *Preprint*, arXiv:2412.15287.

DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning.

Yihe Deng, Hritik Bansal, Fan Yin, Nanyun Peng, Wei Wang, and Kai-Wei Chang. 2025. Openvlthinker: An early exploration to complex vision-language reasoning via iterative self-improvement. *Preprint*, arXiv:2503.17352.

Kanishk Gandhi, Ayush Chakravarthy, Anikait Singh, Nathan Lile, and Noah D. Goodman. 2025. Cognitive behaviors that enable self-improving reasoners, or, four habits of highly effective stars. *Preprint*, arXiv:2503.01307.

Jiahui Gao, Renjie Pi, Jipeng Zhang, Jiacheng Ye, Wanjun Zhong, Yufei Wang, Lanqing Hong, Jianhua Han, Hang Xu, Zhenguo Li, and Lingpeng Kong. 2023. G-llava: Solving geometric problem with multi-modal large language model. *Preprint*, arXiv:2312.11370.

Sachin Goyal, Ziwei Ji, Ankit Singh Rawat, Aditya Krishna Menon, Sanjiv Kumar, and Vaishnavh Nagarajan. 2024. Think before you speak: Training language models with pause tokens. *Preprint*, arXiv:2310.02226.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *Preprint*, arXiv:1503.02531.

Audrey Huang, Adam Block, Dylan J. Foster, Dhruv Rohatgi, Cyril Zhang, Max Simchowitz, Jordan T. Ash, and Akshay Krishnamurthy. 2024. Self-improvement in language models: The sharpening mechanism. *Preprint*, arXiv:2412.01951.

Wenxuan Huang, Bohan Jia, Zijie Zhai, Shaosheng Cao, Zheyu Ye, Fei Zhao, Zhe Xu, Yao Hu, and Shaohui Lin. 2025. Vision-r1: Incentivizing reasoning capability in multimodal large language models. *Preprint*, arXiv:2503.06749.

Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, Lei M Zhang, Kay McKinney, Disha Shrivastava, Cosmin Paduraru, George Tucker, Doina Precup, Feryal Behbahani, and Aleksandra Faust. 2025. Training language models to self-correct via reinforcement learning. In *Proceedings of the 12th International Conference on Learning Representations (ICLR)*.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. *Preprint*, arXiv:2305.20050.

Yuqi Liu, Bohao Peng, Zhisheng Zhong, Zihao Yue, Fanbin Lu, Bei Yu, and Jiaya Jia. 2025. Seg-zero: Reasoning-chain guided segmentation via cognitive reinforcement. *Preprint*, arXiv:2503.06520.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. 2024. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *Preprint*, arXiv:2310.02255.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. *Preprint*, arXiv:2303.17651.

OpenAI. 2024. Openai o1 system card. *Preprint*, arXiv:2412.16720.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *Preprint*, arXiv:2203.02155.

Yuxiao Qu, Tianjun Zhang, Naman Garg, and Aviral Kumar. 2024. Recursive introspection: Teaching language model agents how to self-improve. *Preprint*, arXiv:2407.18219.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36:53728–53741.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *Preprint*, arXiv:2402.03300.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *Preprint*, arXiv:2408.03314.

Yuda Song, Hanlin Zhang, Carson Eisenach, Sham Kakade, Dean Foster, and Udaya Ghai. 2025. Mind the gap: Examining the self-improvement capabilities of large language models. *Preprint*, arXiv:2412.02674.

Haozhe Wang, Chao Qu, Zuming Huang, Wei Chu, Fangzhen Lin, and Wenhu Chen. 2025. Vl-rethinker: Incentivizing self-reflection of vision-language models with reinforcement learning. *Preprint*, arXiv:2504.08837.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Xuezhi Wang and Denny Zhou. 2024. Chain-of-thought reasoning without prompting. In *Advances in Neural Information Processing Systems*, volume 37, pages 66383–66409. Curran Associates, Inc.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. 2025a. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild. *Preprint*, arXiv:2503.18892.

Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. 2025b. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild. *Preprint*, arXiv:2503.18892.

Jingyi Zhang, Jiaxing Huang, Huanjin Yao, Shunyu Liu, Xikun Zhang, Shijian Lu, and Dacheng Tao. 2025. R1-vl: Learning to reason with multimodal large language models via step-wise group relative policy optimization. *Preprint*, arXiv:2503.12937.

Hengguang Zhou, Xirui Li, Ruochen Wang, Minhao Cheng, Tianyi Zhou, and Cho-Jui Hsieh. 2025. R1-zero's "aha moment" in visual reasoning on a 2b non-sft model. *Preprint*, arXiv:2503.05132.

# A Appendix

### Verifier Prompt

Now you act as a judge, helping me determine which of the <length> texts I provide better answers the question.
Question: <question>
Repsonse: <response>
Please strictly follow the following format requirements when outputting, and don't have any other unnecessary words.
Output format: "I choose response [number] because"

### AHA Search Prompt

Below is a chain-of-reasoning generated by a Language Model when attempting to solve a math problem. Evaluate this chain-of-reasoning to determine whether it demonstrates beneficial problem-solving behaviors that deviate from typical linear, monotonic reasoning patterns commonly observed in language models.
<start_of_reasoning> <RESPONSE> <end_of_reasoning>
Specifically, actively identify and emphasize beneficial behaviors such as:
(1) Backtracking: Explicitly revising approaches upon identifying errors or dead ends (e.g., "This approach won't work because...").
(2) Verification: Systematically checking intermediate results or reasoning steps (e.g., "Let's verify this result by...").
Additionally, remain attentive to and encourage the identification of other beneficial behaviors not explicitly listed here, such as creative analogies, abstraction to simpler cases, or insightful generalizations.
Important:
Clearly specify each beneficial behavior you identify.
If there is strong example of this, provide <YES> followed by specific explanations. Otherwise, provide <NO>
A positive response example:
<YES> This contains Backtracking and Verification, respectively from "example quote" and "example quote"
A negative response example, no further explanation is needed at all, SIMPLY return <NO>:
<NO>

### MathVista Parser Prompt

Please read the following example. Then extract the answer from the model response and type it at the end of the prompt.
Hint: Please answer the question requiring an integer answer and provide the final value, e.g., 1, 2, 3, at the end. Question: Which number is missing?
Model response: The number missing in the sequence is 14.
Extracted answer: 14
Hint: Please answer the question requiring a floating-point number with one decimal place and provide the final value, e.g., 1.2, 1.3, 1.4, at the end.
Question: What is the fraction of females facing the camera?
Model response: The fraction of females facing the camera is 0.6, which means that six out of ten females in the group are facing the camera.

Extracted answer: 0.6
Hint: Please answer the question requiring a floating-point number with two decimal places and provide the final value, e.g., 1.23, 1.34, 1.45, at the end. Question: How much money does Luca need to buy a sour apple candy and a butterscotch candy? (Unit: $)
Model response: Luca needs $1.45 to buy a sour apple candy and a butterscotch candy.
Extracted answer: 1.45
Hint: Please answer the question requiring a Python list as an answer and provide the final list, e.g., [1, 2, 3], [1.2, 1.3, 1.4], at the end.
Question: Between which two years does the line graph saw its maximum peak?
Model response: The line graph saw its maximum peak between 2007 and 2008.
Extracted answer: [2007, 2008]
Hint: Please answer the question and provide the correct option letter, e.g., A, B, C, D, at the end. Question: What fraction of the shape is blue?Choices: (A) 3/11 (B) 8/11 (C) 6/11 (D) 3/5
Model response: The correct answer is (B) 8/11.
Extracted answer: B
<query><response>